

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	302	717/106.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:37
L2	250	717/158.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:38
L3	427	717/140.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
L4	11	717/106.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
L5	13	717/118.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
L6	17	717/140.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
L7	8	717/158.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
S1	2	"6321379".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/07/19 08:08
S2	0	"6321379".URPN.	USPAT	OR	OFF	2004/07/19 08:09

S3	33	("4819234"   "4872167"   "5168554"   "5504932"   "5533192"   "5557761"   "5594864"   "5632032"   "5712996"   "5754855"   "5768591"   "5774721"   "5787245"   "5805892"   "5812811"   "5867643"   "5877766"   "5887166"   "5901315"   "5903730"   "5913925"   "5953530"   "5961639"   "5978902"   "6002872"   "6009269"   "6058493"   "6059840"   "6072952"   "6094716"   "6101524"   "6112293"   "6151701").PN.	USPAT	OR	OFF	2004/07/19 08:09
S4	3	("4819234"   "4872167"   "5168554"   "5504932"   "5533192"   "5557761"   "5594864"   "5632032"   "5712996"   "5754855"   "5768591"   "5774721"   "5787245"   "5805892"   "5812811"   "5867643"   "5877766"   "5887166"   "5901315"   "5903730"   "5913925"   "5953530"   "5961639"   "5978902"   "6002872"   "6009269"   "6058493"   "6059840"   "6072952"   "6094716"   "6101524"   "6112293"   "6151701").PN. and (branch adj address)	USPAT	OR	OFF	2004/07/19 08:10
S5	10	("4819234"   "4872167"   "5168554"   "5504932"   "5533192"   "5557761"   "5594864"   "5632032"   "5712996"   "5754855"   "5768591"   "5774721"   "5787245"   "5805892"   "5812811"   "5867643"   "5877766"   "5887166"   "5901315"   "5903730"   "5913925"   "5953530"   "5961639"   "5978902"   "6002872"   "6009269"   "6058493"   "6059840"   "6072952"   "6094716"   "6101524"   "6112293"   "6151701").PN. and (target adj address)	USPAT	OR	OFF	2004/07/19 08:32

S6	79	(architecture or environment or platform) same ( cycles and (instruction near3 issu\$3)) and ( frequency or cost or performance or metric or metadata ) and compil\$5	USPAT	OR	OFF	2004/07/19 10:25
S7	1	"6389529".pn. and (stall\$3 or wait\$3 or miss\$3.)	USPAT	OR	OFF	2004/07/19 10:27
S8	1	"5,202,993".pn. and (environment or architecture or platform)	USPAT	OR	OFF	2004/07/19 11:21
S9	1	"5961637".pn.	USPAT	OR	OFF	2004/07/19 11:26
S10	7	"5961637".URPN.	USPAT	OR	OFF	2004/07/19 11:23
S11	5	rodeh.in. and (branch\$3 or instruction or optimiz\$7 or target )	USPAT	OR	OFF	2004/07/19 11:28
S12	137	717/148.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S13	195	717/106.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S14	295	717/124.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S15	186	717/130.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S16	256	717/140.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S17	188	717/158.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:07
S18	157	(static\$4 near5 compil\$5 ) and (instrument\$5 or hook\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:13

S19	64	(generat\$3 near\$5 instrument\$5) and ((field or attribute or value) near\$5 ( watch or inspect\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:20
S20	51	instrument\$5 near\$3 native	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:15
S21	5670	flag same (redirect\$3 or (instruction adj pointer) or offset or jump)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:19
S22	11	jvmdi	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:22
S23	11	jvmdi and (event or hook or debug or field or flag or watch or state or stack or library or dll or jump or stub or offset or redirect\$3 or instrument\$5 or guard\$3 or predicat\$3 or compil\$5 )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:25
S24	2	(generat\$3 near\$5 instrument\$5) and ((field or attribute or value) near\$5 ( watch or inspect\$3)) and (sav\$3 near\$3 state) and register and event and restor\$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:41
S25	2	flag same (redirect\$3 or (instruction adj pointer) or offset or jump) and (instrument\$5 near\$3 native and (JIT or java or bytecode or byte-code or (byte adj code)) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:41
S26	14	instrument\$5 near\$3 native and (JIT or java or bytecode or byte-code or (byte adj code))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 15:41
S27	10	"6170083".URPN.	USPAT	OR	OFF	2004/01/29 16:23
S28	6	(static\$4 near\$5 compil\$5 ) and (instrument\$5 or hook\$3) and ( (guard\$3 or predicat\$5 or flag\$4 ) near\$5 (watch or inspect\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 16:28

S29	8	(generat\$3 near\$5 instrument\$5) near5 ((field or attribute or value) near5 ( watch or inspect\$3))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 16:28
S30	41	(static\$4 near\$5 compil\$5 ) and (instrument\$5 or hook\$3) and (java or bytecode or byte-code or (byte adj code))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/01/29 16:29
S31	10	("5901315" "6078744" "6298477" "6637024" "6662358" ).pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/19 10:39
S32	2	"20010047510"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/19 10:41
S33	2	"20010047510"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/02/19 11:24
S34	11979	(function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2004/07/14 14:31
S35	11911	(function or procedure or method) near3 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/07/15 11:53
S36	23	"5560009"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/07/15 11:53
S37	2	"5560009".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2004/07/15 11:53
S38	6	(java or bytecode or byte-code or (byte adj code) ) near3 ( (access\$3 or modif\$7) adj field )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/04 16:29

S39	2	(java near3 (memory adj model) ) near3 ( (access\$3 or modif\$7) adj field )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:17
S40	0	java near3 "debug support code"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:18
S41	2	java and "debug support code"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:18
S42	2	"debug support code"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:18
S43	2	jit and "debug support code"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:18
S44	36	jit and debug near3 code	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:19
S45	0	jit and debug near3 (code near3 native)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 07:19
S46	2	(java or bytecode or byte-code or (byte adj code)) near3 ( (selective\$2 or minimal or minimiz\$5 or reduc\$4) near5 debug\$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 11:49
S47	9	(java or bytecode or byte-code or (byte adj code)) near3 ( (flag\$4 or activat\$3 or selective\$2 or minimal or minimiz\$5 or reduc\$4) near5 debug\$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 11:50
S48	0	(java or bytecode or byte-code or (byte adj code)) near3 ( (flag\$4 or activat\$3 or selective\$2 or minimal or minimiz\$5 or reduc\$4) near5 instrumentat\$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 11:51

S49	96	(java or bytecode or byte-code or (byte adj code)) near3 (flag\$4 or activat\$3 or selective\$2 or minimal or minimiz\$5 or reduc\$4) and debug\$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/04/05 11:52
S51	230	717/118.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/22 16:55
S52	427	717/140.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:38
S53	250	717/158.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:38
S54	11	717/106.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
S55	13	717/118.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
S56	17	717/140.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39
S57	8	717/158.ccls. and (function or procedure or method) near5 ((field of value) near3 (access\$3 or modify\$3 or modification or edit\$3 or chang\$3 or alter\$3) )	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/05/23 06:39





USPTO

[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Log out](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **java** and **field access** and **guard**Found **34,525** of **154**Sort results  
by
☒ [Save results to a Binder](#)
[Try an Advanced Search](#)
☒ [Search Tips](#)
[Try this search in The ACM Guide](#)
Display  
results
☐ Open results in a new  
window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☒ ☐1 [Type inference for atomicity](#)

Cormac Flanagan, Stephen N. Freund, Marina Lifshin

January 2005 **Proceedings of the 2005 ACM SIGPLAN international workshop on Types in languages design and implementation**Full text available: [pdf\(168.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Atomicity is a fundamental correctness property in multithreaded programs. This paper presents an algorithm for verifying atomicity via type inference. The underlying type system supports guarded, write-guarded, and unguarded fields, as well as thread-local data, parameterized classes and methods, and protected locks. We describe an implementation of this algorithm for Java and discuss its performance and usability on benchmarks totaling sixty thousand lines of code.

**Keywords:** atomicity, concurrency, reduction, type inference2 [Recompilation for debugging support in a JIT-compiler](#)

Mustafa M. Tikir, Jeffrey K. Hollingsworth, Guei-Yuan Lueh

November 2002 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2002 ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering, Volume 28 Issue 1**Full text available: [pdf\(89.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


A static Java compiler converts Java source code into a verifiably secure and compact architecture-neutral intermediate format, called Java *byte codes*. The Java byte codes can be either interpreted by a Java Virtual Machine or translated into native code by Java Just-In-Time compilers. Static Java compilers embed debug information in the Java class files to be used by the source level debuggers. However, the debug information is generated for architecture independent byte codes and most o ...

**Keywords:** Java, Java virtual machine debugger interface, debug information, dynamic recompilation, field access watch, just-in-time compilation



**3** Type-safe multithreading in cyclone


Dan Grossman

January 2003 **ACM SIGPLAN Notices , Proceedings of the 2003 ACM SIGPLAN international workshop on Types in languages design and implementation, Volume 38 Issue 3**Full text available:  [pdf\(228.33 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We extend Cyclone, a type-safe polymorphic language at the C level of abstraction, with threads and locks. Data races can violate type safety in Cyclone. An extended type system statically guarantees their absence by enforcing that thread-shared data is protected via locking and that thread-local data does not escape the thread that creates it. The extensions interact smoothly with parametric polymorphism and region-based memory management. We present a formal abstract machine that models the ne ...

**Keywords:** cyclone, data races, types**4** Core semantics of multithreaded Java


Jeremy Manson, William Pugh

June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**Full text available:  [pdf\(816.27 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java has integrated multithreading to a far greater extent than most programming languages. It is also one of the only languages that specifies and requires safety guarantees for improperly synchronized programs. It turns out that understanding these issues is far more subtle and difficult than was previously thought. The existing specification makes guarantees that prohibit standard and proposed compiler optimizations; it also omits guarantees that are necessary for safe execution of much ex ...

**5** Onward!: Finding bugs is easy

David Hovemeyer, William Pugh

October 2004 **Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications**Full text available:  [pdf\(104.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Many techniques have been developed over the years to automatically find bugs in software. Often, these techniques rely on formal methods and sophisticated program analysis. While these techniques are valuable, they can be difficult to apply, and they aren't always effective in finding real bugs.

*<i>Bug patterns</i>* are code idioms that are often errors. We have implemented automatic detectors for a variety of bug patterns found in Java programs. In this extended abstract<s ...

**Keywords:** bug checkers, bug patterns, bugs, static analysis**6** Efficient and precise datarace detection for multithreaded object-oriented programs

Jong-Deok Choi, Keunwoo Lee, Alexey Loginov, Robert O'Callahan, Vivek Sarkar, Manu Sridharan

May 2002 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2002 Conference on Programming language design and implementation**, Volume 37 Issue 5

Full text available:  [pdf\(171.13 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


We present a novel approach to dynamic datarace detection for multithreaded object-oriented programs. Past techniques for on-the-fly datarace detection either sacrificed precision for performance, leading to many false positive datarace reports, or maintained precision but incurred significant overheads in the range of 3x to 30x. In contrast, our approach results in very few false positives and runtime overhead in the 13% to 42% range, making it both efficient *and* precis ...

**Keywords:** dataraces, debugging, multithreaded programming, object-oriented programming, parallel programs, race conditions, static-dynamic co-analysis, synchronization

7 Constructing compact models of concurrent Java programs

James C. Corbett

March 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 1998 ACM SIGSOFT international symposium on Software testing and analysis**, Volume 23 Issue 2

Full text available:  [pdf\(1.06 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Finite-state verification technology (e.g., model checking) provides a powerful means to detect concurrency errors, which are often subtle and difficult to reproduce. Nevertheless, widespread use of this technology by developers is unlikely until tools provide automated support for extracting the required finite-state models directly from program source. In this paper, we explore the extraction of compact concurrency models from Java code. In particular, we show how static pointer analysis, whic ...

**Keywords:** finite-state verification, model extraction, static analysis

8 A type and effect system for atomicity

Cormac Flanagan, Shaz Qadeer

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available:  [pdf\(266.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Ensuring the correctness of multithreaded programs is difficult, due to the potential for unexpected and nondeterministic interactions between threads. Previous work addressed this problem by devising tools for detecting *race conditions*, a situation where two threads simultaneously access the same data variable, and at least one of the accesses is a write. However, verifying the absence of such simultaneous-access race conditions is neither necessary nor sufficient to ensure the absence o ...

**Keywords:** atomicity, multithreading, race conditions, static checking

9 Technical papers: concurrency: Assuring and evolving concurrent programs: annotations and policy

Aaron Greenhouse, William L. Scherlis

May 2002 **Proceedings of the 24th International Conference on Software Engineering**


Full text available:  [pdf\(1.38 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Assuring and evolving concurrent programs requires understanding the concurrency-related design decisions used in their implementation. In Java-style shared-memory programs, these decisions include which state is shared, how access to it is regulated, the roles of threads, and the policy that distinguishes desired concurrency from race conditions. These decisions rarely have purely local manifestations in code. In this paper, we use case studies from production Java code to explore the costs and ...

#### 10 [Ada, C, C++, and Java vs. the Steelman](#)

David A. Wheeler

July 1997 **ACM SIGAda Ada Letters**, Volume XVII Issue 4


Full text available:  [pdf\(1.57 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper compares four computer programming languages (Ada95, C, C++, and Java) with the requirements of "Steelman", the original 1978 requirements document for the Ada computer programming language. This paper provides a view of the capabilities of each of these languages, and should help those trying to understand their technical similarities, differences, and capabilities.

#### 11 [Zones, contracts and absorbing changes: an approach to software evolution](#)

Huw Evans, Peter Dickman

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 34 Issue 10


Full text available:  [pdf\(2.46 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes a novel approach to managing the evolution of distributed, persistent systems at run-time. This is achieved by partitioning a system into disjoint zones, each of which can be evolved without affecting code in any other. Contracts are defined between zones, making type-level interdependencies and inter-zone communication explicit. Programmer supplied code is added to the running system, at the boundary between zones, to constrain the sco ...

#### 12 [Using shape analysis to reduce finite-state models of concurrent Java programs](#)

James C. Corbett

January 2000 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 9 Issue 1

Full text available:  [pdf\(284.92 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Finite-state verification (e.g., model checking) provides a powerful means to detect concurrency errors, which are often subtle and difficult to reproduce. Nevertheless, widespread use of this technology by developers is unlikely until tools provide automated support for extracting the required finite-state models directly from program source. Unfortunately, the dynamic features of modern languages such as Java complicate the construction of compact finite-state models for verification. I ...

**Keywords:** Java, concurrent systems, finite-state verification, model extraction, modeling, shape analysis, state-space reductions

**13** Cloning-based context-sensitive pointer alias analysis using binary decision diagrams

John Whaley, Monica S. Lam

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation**, Volume 39 Issue 6

Full text available:  [pdf\(277.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This paper presents the first scalable context-sensitive, inclusion-based pointer alias analysis for Java programs. Our approach to context sensitivity is to create a clone of a method for every context of interest, and run a *context-insensitive* algorithm over the expanded call graph to get *context-sensitive* results. For precision, we generate a clone for every acyclic path through a program's call graph, treating methods in a strongly connected component as a single node. Normally ...

**Keywords:** Datalog, Java, binary decision diagrams, cloning, context-sensitive, inclusion-based, logic programming, pointer analysis, program analysis, scalable

**14** Improving the Java memory model using CRF

Jan-Willem Maessen, Xiaowei Shen

October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 35 Issue 10

Full text available:  [pdf\(193.84 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


This paper describes alternative memory semantics for Java programs using an enriched version of the Commit/Reconcile/Fence (CRF) memory model [16]. It outlines a set of reasonable practices for safe multithreaded programming in Java. Our semantics allow a number of optimizations such as load reordering that are currently prohibited. Simple thread-local algebraic rules express the effects of optimizations at the source or bytecode level. The rules focus on reordering source-level operations; the ...

**Keywords:** Java, commit/reconcile/fence, compilation, memory models

**15** Implementing jalapeño in Java

Bowen Alpern, C. R. Attanasio, Anthony Cocchi, Derek Lieber, Stephen Smith, Ton Ngo, John J. Barton, Susan Flynn Hummel, Janice C. Sheperd, Mark Mergen

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 34 Issue 10

Full text available:  [pdf\(1.57 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Jalapeño is a virtual machine for Java™ servers written in Java. A running Java program involves four layers of functionality: the user code, the virtual-machine, the operating system, and the

hardware. By drawing the Java / non-Java boundary below the virtual machine rather than above it, Jalapeño reduces the boundary-crossing overhead and opens up more opportunities for optimization. To get Jalapeño started, a boot image of a ...

#### 16 Practical predicate dispatch

Todd Millstein

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**, Volume 39 Issue 10

Full text available:  pdf(192.02 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

*<i>*Predicate dispatch*</i>* is an object-oriented (OO) language mechanism for determining the method implementation to be invoked upon a message send. With predicate dispatch, each method implementation includes a predicate guard specifying the conditions under which the method should be invoked, and logical implication of predicates determines the method overriding relation. Predicate dispatch naturally unifies and generalizes several common forms of dynamic dispatch, including tradi ...

**Keywords:** dynamic dispatch, modular typechecking, predicate dispatch

#### 17 Technical papers: program analysis: Specifying multithreaded Java semantics for program verification

Abhik Roychoudhury, Tulika Mitra

May 2002 **Proceedings of the 24th International Conference on Software Engineering**


Full text available:  pdf(1.27 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Java programming language supports multithreading where the threads interact among themselves via read/write of shared data. Most current work on multithreaded Java program verification assumes a model of execution that is based on interleaving of the operations of the individual threads. However, the Java language specification (which any implementations of Java multithreading must follow) supports a weaker model of execution, called the Java Memory Model (JMM). The JMM allows certain reord ...

#### 18 New techniques for security and reliability enhancement in embedded systems: Analyzing heap error behavior in embedded JVM environments

G. Chen, M. Kandemir, N. Vijaykrishnan, A. Sivasubramaniam, M. J. Irwin

September 2004 **Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis**

Full text available:  pdf(285.76 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Recent studies have shown that transient hardware errors caused by external factors such as alpha particles and cosmic ray strikes can be responsible for a large percentage of system down-time. Denser processing technologies, increasing clock speeds, and low supply voltages used in embedded systems can worsen this problem. In many embedded environments, one may not want to provision extensive error protection in hardware because of (i) form-factor or power consumption limitations, and/or (ii) to ...

**Keywords:** JVM, softerrors

**19** Type-based race detection for Java

Cormac Flanagan, Stephen N. Freund

May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5


Full text available:  [pdf\(237.37 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a static race detection analysis for multithreaded Java programs. Our analysis is based on a formal type system that is capable of capturing many common synchronization patterns. These patterns include classes with internal synchronization, classes that require client-side synchronization, and thread-local classes. Experience checking over 40,000 lines of Java code with the type system demonstrates that it is an effective approach for eliminating races conditions. On lar ...

**20** Stack allocation and synchronization optimizations for Java using escape analysis

Jong-Deok Choi, Manish Gupta, Mauricio J. Serrano, Vugranam C. Sreedhar, Samuel P. Midkiff

November 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 25 Issue 6

Full text available:  [pdf\(632.85 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article presents an *escape analysis* framework for Java to determine (1) if an object is not reachable after its method of creation returns, allowing the object to be allocated on the stack, and (2) if an object is reachable only from a single thread during its lifetime, allowing unnecessary synchronization operations on that object to be removed. We introduce a new program abstraction for escape analysis, the *connection graph*, that is used to establish reachability relationships ...





**Keywords:** Connection graphs, escape analysis, points-to graph

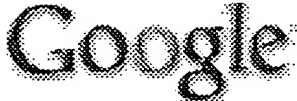
Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Playe](#)


[Web](#) [Images](#) [Groups](#) [News](#) [Froogle](#) [Local](#) [more »](#)  
  [Advanced Search](#)  
[Preferences](#)

Web

Results 1 - 10 of about 5,580,000 for **java + field access** . (0.23 seconds)SQL-J Language Reference

... NEW: **Field access** works with the new class alias construct. ... NEW: In prior releases, if an expression referenced a **Java field** that was null, ...

www.cs.nott.ac.uk/TSG/manuals/java/JDK12EE/cloudscape/doc/html/coredocs/sqlj113.htm - 9k - [Cached](#) - [Similar pages](#)

Accessing Java Fields

... The JNI provides functions that native methods use to get and set **Java fields**.

... to accessing **Java** methods, you use different JNI functions to **access** ...

www.science.uva.nl/ict/ossdocs/java/tutorial/native1.1/implementing/field.html - 6k - [Cached](#) - [Similar pages](#)

Implementing Java Reflection Page

... This is fine, because reflection is supposed to preserve semantics of **Java**

**field access**, etc. However, the current reflection API does specify that in ...

cs-www.cs.yale.edu/homes/hamid/Java\_Reflection/implrefl.html - 12k - [Cached](#) - [Similar pages](#)

Returning a Java Class from a Method Call or Field Access

... A **Java** object returned from a **Java** method call or **field access** in a script is automatically wrapped in its own ChiliBeans wrapper. ...

docs.sun.com/source/817-2514-10/Ch12\_ChiliBean11.html - 8k - [Cached](#) - [Similar pages](#)

15.11.1 Field Access Using a Primary

The **Java** Spec Report is no longer maintained. The author disclaims any ...

Otherwise, the result of the **field access** expression is computed as follows: ...

www.ergnosis.com/java-spec-report/java-language/jls-15.11.1.html - 8k - [Cached](#) - [Similar pages](#)

JRI Enhancements to javah

... it defines macros to call methods and **access fields** of objects ... In the case where the **Java field** is static, the ClassName\* self parameter will be ...

home.netscape.com/eng/jri/jri-javah.html - 25k - May 21, 2005 - [Cached](#) - [Similar pages](#)

Java Tip 126: Prepare cross-server database access methods with JDBC

... You can retrieve the query result data **fields** as **Java** objects from ...

Class AccessMethods (see AccessMethods.java) puts together the database **access** ...

www.javaworld.com/javaworld/javatips/jw-javatip126.html - [Similar pages](#)

A Primer for Aspect-Oriented Programming in Java

... augmented, or redirected, as can **field access** and even inheritance—in many

... Other **Java** AOP implementations use keywords and extensions to the **Java** ...

www.informit.com/articles/article.asp?p=174533 - 19k - [Cached](#) - [Similar pages](#)

Pnuts User's Guide

... The behavior of **field access**, **index access**, and method/constructor call, ...

in pnuts.lang package, each of which defines a behavior of **Java** API **access**. ...



[www-106.ibm.com/developerworks/java/library/j-dyn0603/](http://www-106.ibm.com/developerworks/java/library/j-dyn0603/) - 61k - Cached - Similar pages

Googooooooooooooole ►

Result Page:    1   2   3   4   5   6   7   8   9   10    Next

**Free! Google Desktop Search: Search your own computer. Download now.**

**Find:**  emails -  files -  chats -  web history -  media -  PDF

## java + field access

## Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2005 Google



Home | Login | Logout | Access Information  
Site

Welcome United States Patent and Trademark  
Office

## Search Results

BROWSE

SEARCH

IEEE XPLORE  
GUIDE

Results for "(( java<in>metadata ) <and> ( field<in>metadata ) <and> ( access<in>m..."

e-mail

Your search matched 20 of 1160635 documents.

A maximum of 100 results are displayed, 25 to a page, sorted by **Relevance** in **Descending** order.

» [View Session](#)

History

» [New Search](#)

Modify Search

(( java<in>metadata ) <and> ( field<in>metadata ) <and> ( access<in>metadata

» Key

☐ Check to search only within this results set

IEEE  
JNL

IEEE  
Journal or  
Magazine

Display  
Format:

☒ Citation ☐ Citation & Abstract

IEEE  
JNL

IEEE Journal  
or  
Magazine

Select Article Information

IEEE  
CNF

IEEE  
Conference  
Proceeding

- ☐ 1. **The Java CAN API-a Java gateway to field bus communication**  
Buhler, D.; Nusser, G.;  
Factory Communication Systems, 2000. Proceedings. 2000 IEEE  
Workshop on  
6-8 Sept. 2000 Page(s):37 - 43  
AbstractPlus | Full Text: PDF(556 KB) IEEE CNF

IEEE  
CNF

IEEE  
Conference  
Proceeding

- ☐ 2. **The real experiment eXecution approach to networking course**  
Chirico, M.; Giudici, F.; Sappia, A.; Scapolla, A.M.;  
Education, IEEE Transactions on  
Volume 40, Issue 4, Nov. 1997 Page(s):4 pp.  
AbstractPlus | Full Text: PDF(16 KB) IEEE JNL

IEEE  
STD

IEEE  
Standard

- ☐ 3. **The ENVISAT data products**  
Levrini, G.; Brooker, G.;  
Geoscience and Remote Sensing Symposium, 2000. Proceedings.  
2000. IEEE 2000 International  
Volume 3, 24-28 July 2000 Page(s):1198 - 1201 vol.3  
AbstractPlus | Full Text: PDF(320 KB) IEEE CNF

- ☐ 4. **Managing scientific metadata**  
Jones, M.B.; Berkley, C.; Bojilova, J.; Schildhauer, M.;  
Internet Computing, IEEE  
Volume 5, Issue 5, Sept.-Oct. 2001 Page(s):59 - 68  
AbstractPlus | References | Full Text: PDF(240 KB) IEEE JNL

- ☐ 5. **Internet-based learning by doing**  
Anido, L.; Llamas, M.; Fernandez, M.J.;  
Education, IEEE Transactions on  
Volume 44, Issue 2, May 2001 Page(s):18 pp.  
AbstractPlus | Full Text: PDF(8 KB) IEEE JNL

- ☐ **6. User interface technologies for home appliances and networks**  
Corcoran, P.M.; Papal, F.; Zoldi, A.;  
Consumer Electronics, IEEE Transactions on  
Volume 44, Issue 3, Aug. 1998 Page(s):679 - 685  
AbstractPlus | References | Full Text: PDF(928 KB) IEEE JNL
- ☐ **7. Supporting object accesses in a Java processor**  
Vijaykrishnan, N.; Ranganathan, N.;  
Computers and Digital Techniques, IEE Proceedings-  
Volume 147, Issue 6, Nov. 2000 Page(s):435 - 443  
AbstractPlus | Full Text: PDF(848 KB) IEE JNL
- ☐ **8. World Wide Web access to real-time and historical data from array of moored buoys in the tropical Pacific Ocean: updates I**  
Soreide, N.N.; McClurg, D.C.; Zhu, W.H.; McPhaden, M.J.; Denb  
Renton, M.W.;  
OCEANS '96. MTS/IEEE. 'Prospects for the 21st Century'. Confe  
Proceedings  
Volume 3, 23-26 Sept. 1996 Page(s):1354 - 1359 vol.3  
AbstractPlus | Full Text: PDF(1472 KB) IEEE CNF
- ☐ **9. The Weather Visualizer: a Java tool for interactive learning**  
Hall, S.E.; Ramamurthy, M.K.; Wilhelmson, R.B.; Plutchak, J.; W  
Sridhar, M.;  
Geoscience and Remote Sensing Symposium, 1996. IGARSS '96.  
Sensing for a Sustainable Future.', International  
Volume 3, 27-31 May 1996 Page(s):1498 - 1500 vol.3  
AbstractPlus | Full Text: PDF(412 KB) IEEE CNF
- ☐ **10. Briki: an optimizing Java compiler**  
Cierniak, M.; Wei Li;  
Compcon '97. Proceedings, IEEE  
23-26 Feb. 1997 Page(s):179 - 184  
AbstractPlus | Full Text: PDF(480 KB) IEEE CNF
- ☐ **11. HIGHROBOT: telerobotics in the Internet**  
Kuchlin, W.; Gruhler, G.; Lumpp, Th.; Speck, A.; Rupp, A.;  
Emerging Technologies and Factory Automation Proceedings, 19  
'97., 1997 6th International Conference on  
9-12 Sept. 1997 Page(s):115 - 120  
AbstractPlus | Full Text: PDF(552 KB) IEEE CNF
- ☐ **12. GeoBrowse: an integrated environment for satellite image ret  
mining**  
Marchisio, G.B.; Wen-Hao Li; Sannella, M.; Goldschneider, J.R.;  
Geoscience and Remote Sensing Symposium Proceedings, 1998.  
1998 IEEE International  
Volume 2, 6-10 July 1998 Page(s):669 - 673 vol.2  
AbstractPlus | Full Text: PDF(1496 KB) IEEE CNF
- ☐ **13. Virtual Java devices. Integration of fieldbus based systems in**  
Lumpp, T.; Gruhler, G.; Kuchlin, W.;  
Industrial Electronics Society, 1998. IECON '98. Proceedings of t  
Annual Conference of the IEEE  
Volume 1, 31 Aug.-4 Sept. 1998 Page(s):176 - 181 vol.1  
AbstractPlus | Full Text: PDF(692 KB) IEEE CNF
- 14. Method execution on a distributed image processing back-end**

- ☐ Niederl, F.; Goller, A.;  
Parallel and Distributed Processing, 1998. PDP '98. Proceedings of the  
Euromicro Workshop on  
21-23 Jan. 1998 Page(s):243 - 249  
AbstractPlus | Full Text: PDF(664 KB) IEEE CNF
- ☐ 15. **Linux in factory automation? Internet controlling of fieldbus**  
Kastner, W.; Csebits, C.; Mayer, M.;  
Emerging Technologies and Factory Automation, 1999. Proceedings  
'99. 1999 7th IEEE International Conference on  
Volume 1, 18-21 Oct. 1999 Page(s):27 - 31 vol.1  
AbstractPlus | Full Text: PDF(536 KB) IEEE CNF
- ☐ 16. **Low cost wireless LAN based medical informatics system**  
Pedersen, P.C.; Pahlavan, K.; Auk, S.; Sullivan, W.; Nguyen, T.;  
[Engineering in Medicine and Biology, 1999. 21st Annual Conf. &  
Annual Fall Meeting of the Biomedical Engineering Soc.] BMES  
Conference, 1999. Proceedings of the First Joint  
Volume 2, 13-16 Oct. 1999 Page(s):1225 vol.2  
AbstractPlus | Full Text: PDF(80 KB) IEEE CNF
- ☐ 17. **Macro-based hardware compilation of Java<sup>TM</sup> bytecodes into reconfigurable computing system**  
Cardoso, J.M.P.; Neto, H.C.;  
Field-Programmable Custom Computing Machines, 1999. FCCM  
Proceedings. Seventh Annual IEEE Symposium on  
21-23 April 1999 Page(s):2 - 11  
AbstractPlus | Full Text: PDF(812 KB) IEEE CNF
- ☐ 18. **Remote fieldbus system management with Java and XML**  
Buhler, D.; Kuchlin, W.;  
Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000  
International Symposium on  
Volume 1, 4-8 Dec. 2000 Page(s):1 - 6 vol.1  
AbstractPlus | Full Text: PDF(848 KB) IEEE CNF
- ☐ 19. **Internet access to Great Lakes CoastWatch remote sensing in**  
Leshkevich, G.A.; Liu, S.;  
Geoscience and Remote Sensing Symposium, 2000. Proceedings.  
2000. IEEE 2000 International  
Volume 5, 24-28 July 2000 Page(s):2077 - 2079 vol.5  
AbstractPlus | Full Text: PDF(416 KB) IEEE CNF
- ☐ 20. **On integrating mobile devices into a workflow management system**  
Muller-Wilken, S.; Wienberg, F.; Lamersdorf, W.;  
Database and Expert Systems Applications, 2000. Proceedings. 11th  
International Workshop on  
4-8 Sept. 2000 Page(s):186 - 190  
AbstractPlus | Full Text: PDF(476 KB) IEEE CNF



Help Contact Us  
Security

Indexed by  
**InsPEC**

© Copyright 2000  
IEEE